From: "Miles Smid" <masmid@erols.com>
To: <aesround2@nist.gov>
Cc: <jfoti@nist.gov>, <edward.roback@nist.gov>, <william.burr@nist.gov>,
"Miles Smid" <smid@cygnacom.com>
Subject: AES Round 2 Comments
Date: Tue, 23 May 2000 17:44:41 -0400
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook IMO, Build 9.0.2416 (9.0.2910.0)
Importance: Normal
X-MimeOLE: Produced By Microsoft MimeOLE V5.00.2615.200

Dear Sir:

Please see the three attachments containing my AES Round 2 Comments.

Thank you very much,

Miles Smid

May 22, 2000

AES Issues

Miles E. Smid
CygnaCom Solutions

Introduction:

This report was developed for the National Institute of Standards and Technology (NIST) by the author as the second deliverable under Purchase Order 43SBNB067018. It is intended to summarize the key issues discussed at the AES3 conference held in New York City on April 13-14, 2000 and other issues that still need to be addressed. Issues are grouped according to evaluation factors and unresolved issues are indicated by an "*". This paper supplements [1], the first deliverable under the cited purchase order. The opinions expressed in this paper are those of the author and are not intended to reflect any positions held by the NIST AES team or official positions of CygnaCom Solutions.

Security:

1. **Do all candidate algorithms offer an adequate security margin?***
While Serpent, Twofish, and MARS appear to have large security margins, the adequacy of the security margins for RC6 and Rijndael have been called into question. Some of the AES3 conference attendees said that they favored the selection of Rijndael with extra rounds. While a low security margin does not mean that an algorithm is insecure and a high margin does not mean that the algorithm is secure, the security margin seems to be the best shorthand way of describing the current state of knowledge about an algorithm's security. Cryptographers attack symmetric-key algorithms by first attacking reduced round variants and then working their way up to the full algorithm. Some increases in the number of rounds for the best attack on certain algorithms were presented at the AES3 conference and it is likely that additional increases will be made. One would think that any algorithm proposed for the AES should have a comfortable security margin. On the other hand, the submitters of RC6 and Rijndael claim that their security margins are adequate. This issue remains unresolved. NIST has the option of adding rounds to an otherwise excellent algorithm if it feels that such a modification would result in the best standard.

2. **Should extra rounds be added to winning algorithms with low security margins?***
The most discussed case is Rijndael-128 where there is an attack on seven rounds. Some have questioned whether only three additional rounds offers a sufficient security margin. Nevertheless, Rijndael compares very favorably to its competitors in most other areas. It has been suggested that NIST should publish Rijndael with increased rounds as the standard. Some thought that this

modification would be superior to having to increase the rounds at a later date when cryptanalysis might show ten rounds to be inadequate. The ultimate decision on this issue will be up to NIST.

3. **What gives us confidence in the security of algorithms?***
   The submitters of RC6 argued in their submitter presentation that security margin was not necessarily the most important factor in determining an algorithm's security. They argued that the amount of scrutiny that an algorithm receives is more significant. Security margins on the other hand benefit the algorithms that have been studied the least. "The true security of a cipher depends on
   - the amount of cryptanalytic scrutiny received,
   - the accuracy of existing cryptanalysis,
   - the ease with which verifying experiments can be conducted on a cipher,
   - the amount of earlier cryptanalytic work that can be used in the assessment of the cipher, and
   - the accuracy of the designers initial estimates" [2].

   Based on these factors, they feel that RC6, which is simple and based on the well known RC5, offers excellent security. This issue is and will remained unresolved but it is likely that NIST will consider these factors along with security margin and perhaps other factors to make its decision on the security provided by the candidates.

4. **How much weight should be given to timing and power analysis attacks?**
   Power analysis attacks were a significant topic at AES2 but were not discussed at AES3. There was one paper at FSE2000 that claimed that MARS was the most difficult finalist to mask [3]. While certain operations lend themselves to power analysis attacks, it appears that all the finalists are susceptible to these attacks to some degree. The means to prevent these attacks is now a topic of research. The lack of discussion on this topic at AES3 might be indicative of a general belief that power analysis is not a major factor in distinguishing which algorithms should be selected for the standard.

5. **Is the current security analysis sufficient?**
   It is likely that the majority of the attendees at the conference would say that the current security analysis is not sufficient. Cryptographers are a conservative lot, and it takes years of study to gain confidence in an algorithm. Since fifteen algorithms were originally considered, the efforts of the academic community were diluted and there is a feeling that more study is needed. Hopefully, NIST will get some confirmation from NSA that its selection appears to be secure. In addition the winner(s) will get additional scrutiny during and after the Federal Information Processing Standard (FIPS) development period. Finally, the standard will be reviewed every five years. The general feeling seems to be that more study is needed but that the process should continue as planned.

6. **Are all candidate algorithms sufficiently secure?***
   All the AES finalists appear to offer very high security. Even the "attacks" on many reduced round versions are not practical. Since all the finalists appear sufficiently secure, it may make little sense to try to distinguish between them based on subtle security differences. Perhaps cost and flexibility should be the deciding factors. On the other hand, the submitters of Serpent have argued just the opposite. They stated that Serpent meets all of the NIST cost and flexibility requirements and should be selected because it is the most secure. The AES has the potential to be around for many years. Therefore, it should be as secure as possible while still meeting the other NIST requirements. Since all the candidates appear to meet all NIST requirements, NIST will have to decide whether extra security, less cost, or extra flexibility is most important.

7. **What should NIST say about the algorithms not selected?**
   There were comments at the conference that NIST should say something about the security of the algorithms not selected. This would be a reward to those who submitted strong algorithms that were not quite as efficient or flexible as the selected winner(s). While this would be a nice reward for all the work involved in developing and defending a submission, it may not be feasible for NIST to take such an action. Traditionally, NIST does not endorse cryptographic algorithms unless they are specified in, or referenced by, a FIPS. By saying that an algorithm not related to a FIPS was secure, NIST might take on the responsibility to provide for the continuous study of that algorithm. If NIST later changed its assessment of the security of the algorithm, it would need to deal with the owner who might not accept the assessment. In addition, NIST may be obligated to evaluate other algorithms since it should not favor just AES candidates. It seems unlikely that NIST could make security statements about the candidates that are not selected for the AES (or its backup). However, it might be possible for the Secretary of Commerce to send the finalists a letter thanking them for their participation.

Cost:

1. **What is the best approach to minimizing the intellectual property attack on the AES?***
   It was mentioned at the conference that NIST was doing a patent search on the AES finalists. It is assumed that NIST will do an even more extensive search on the selected winner(s). NIST has also published requests for those with intellectual property claims on the algorithms to make them known. Hitachi has responded to NIST citing patents that might be infringed by four of the five finalists. NIST will have to examine the cited patents and claims in order to make a determination of the likelihood of infringement. In view of the fact that the submitters of the algorithms are not charging royalties, NIST should encourage Hitachi to make its license royalty free. If the claim of infringement appears to be valid and there is a license fee, then NIST will have to take the cost into consideration when considering the cost of its selection. There would likely be

strong opposition to the selection of any algorithm which required payment of a license fee since such a fee is counter to the stated AES goals.

It is possible that a party may claim infringement after NIST makes the AES selection. If this happens after the selection is announced but before the AES becomes a FIPS, NIST could change its selection. If it happens after the AES becomes a FIPS, there may be some question as to why the party remained quite for so long a period. Even though two algorithms are more likely to have a claim against them than one, the selection of a winner and a back-up which would be used only if NIST announced a flaw or intellectual property claim on the winner could help to minimize such attacks. If the valid claim were against the backup, then it would never be used, on the other hand, if the valid claim were against the winner, then the backup could be quickly put into use. While there is no perfect defense against an intellectual property claim, frivolous claims will likely incur the wrath of the cryptographic community. One can also hope that the likelihood of such claims will diminish over time.

2. **What is the best approach to minimizing intellectual property claims on efficient implementations?***
   It has been noted that NIST has not restricted the option of charging licensing fees for efficient implementations. This means that the submitter of a winning algorithm or some other party could have patents on all sorts of efficiency improvements to the winning algorithm. This could box implementers in to either using an inefficient implementation or paying a license fee. This is a gray area because NIST felt that it should allow for the licensing of innovative implementations. The difference between an innovation and an extension that would be made by anyone skilled in the art is not always clear. When NIST does its patent search on the winner(s), it may become suspicious if one company holds many patents that constitute improvements on the selected algorithm(s). NIST will have to determine whether these improvements are genuine or whether they are an attempt to box in implementers.

3. **What is the relative importance of hardware versus software performance?**
   AES2 focused mostly on the efficiency of software implementations. At AES3 results were presented that showed that an algorithm could be relatively inefficient in software and efficient in hardware. If multiple algorithms are selected, then NIST might be able to pick the algorithm which maximizes each case. But if one algorithm is selected as the standard, it must operate efficiently in both environments. Some have noted that technology will improve, but there will always be applications that require the most efficient performance possible.

4. **What about memory?**
   Memory like computation speed is an important factor. Several of the papers divided speed by memory when comparing algorithms.

5. **What is the importance of key agility?**
   It was felt that key agility was most important in high speed network applications which implemented the cryptography in hardware. The NSA hardware results seemed to indicate that certain finalists had superior key agility. See [1] for more on how speed, memory, and key agility might be compared.

6. **How significant are Java code results?**
   There was a difference of opinion as to the significance of Java efficiency tests. Some thought that either Java was not a good measure of efficiency or that Java would not be used in applications requiring efficiency. Others felt that Java would be used in applications requiring efficiency or that Java was a good measure of efficiency because of its machine independence. Java timings will probably not be used as the main measure of software efficiency, but merely regarded as another measure of some interest.

7. **Can one draw any general conclusions about efficiency?**
   From the results presented at AES2 and AES3, it appears that Rijndael performs consistently well. RC6 is fast on SGI and PCs but slow on tested Sun systems.

   In FPGA and hardware Rijndael and Serpent appear to do encryption most efficiently followed by Twofish and RC6 with MARS the least efficient. In FPGA and hardware area usage Twofish and RC6 seem to do well. However, since only a few FPGA and hardware implementations were tested, one should not draw strong conclusions from these results.

Algorithm and Implementation Characteristics:

1. **How important are low-end smart card implementations?**
   There was much debate about the importance of low-end smart card applications at AES2 but not much discussion at AES3. Nevertheless, it appears that algorithms which may be implemented in low-end smart cards deserve to receive flexibility credit over those that do not.

2. **How important is on-the-fly key computation?**
   On-the-fly key computation should be a plus in certain applications.

3. **Which is better simplicity or complexity?**
   Quite a bit of discussion centered around complexity versus simplicity at AES3. MARS and Twofish were accused of being complex while RC6, Rijndael, and Serpent were often called simple. The authors of MARS claimed that it was not complex and that it was wise to use different rounds and different operations. The authors of Twofish claimed that its design gave flexibility to optimize the algorithm to the application. The authors of RC6 and Rijndael claimed that the simplicity of their algorithms aided in analysis and efficient implementation. From the AES3 comment forms, it appears that Rijndael won the argument. Mars

and Twofish were cited as being too complex. RC6 seems to have been thought to be designed primarily for 32-bit processors (even though it was the fastest finalist on the 64-bit SGI according to NIST).

4. **Which algorithm(s) are most flexible?**
   A good case can be made for Rijndael. It tends to score among the top two in most categories. It runs well on 8, 32, and 64-bit computers, it gives good performance in FPGA and hardware, it has key agility, and it can be implemented on smart cards. Its memory or area requirements sometimes are a little large possibly because its decryption algorithm is different from its encryption algorithm. When the submitters were asked which algorithm they would select if not their own, two or three said either Rijndael or Rijndael with extra rounds. This again leads to the question as to whether the number of rounds in Rijndael is adequate.

   A case can also be made for Twofish but it doesn't seem to do as well as Rijndael in FPGA and Hardware encryption speed. Serpent does not do as well as the others on many software platforms. RC6 and MARS do not seem to encrypt as efficiently as many other algorithms in FPGA and hardware.

Miscellaneous:

1. **Which algorithms definitely should be selected for the standard?**
   The AES attendees appear to be clearly in favor of Rijndael (86) with Serpent (59) second and Twofish (32) third. Rijndael is perceived to have the fewest performance and implementation weaknesses. Serpent is perceived to be the most secure of the algorithms and is therefore favored by cryptographers either as the winner or as a backup. Twofish is also thought to be flexible and strong but does not have as strong a following as the top two picks. Perhaps most surprising is the weak support given to RC6 (23) considering that it received the most positive votes at the AES2 conference. This seemed to come from the view that RC6 is too oriented to a 32-bit multiply, has a slow key setup, and is patented.

2. **Which algorithms definitely should not be selected for the standard?**
   MARS received the most negative votes (83) followed by RC6 (37) and Twofish (21). In spite of the IBM arguments to the contrary, MARS was perceived to be complex and inflexible. Although it performed well on certain 32-bit processors, it gave lackluster results on SPARC and SUN processors. Finally, its capability to be implemented on low-end smart cards was questioned. In hardware it appeared to follow the pack in speed and memory. In fact, some of the FGA and hardware testers did not even present results for MARS because they thought it unlikely to excel. Twofish also seemed to suffer from the perception of complexity.

   If one subtracts the negative votes from the positive votes, then the following results are obtained.

Rijndael	86-10 = 76
Serpent	59-7 = 52
Twofish	31-21 = 10
RC6	23-37 = -14
MARS	13-83 = -70

It would be difficult to select an algorithm as an AES winner that received a negative score in voting at the AES3 conference.

3. **How many algorithms should NIST select for the AES standard?**
The vast majority (110) of those responding to the question felt that only one algorithm should be selected as a standard. The reasons cited were simplicity, interoperability, and ease of implementation. Eighteen (18) voted for one algorithm plus a backup that would only be used if the standard was found unsuitable for some reason. Only a few (14) wanted two or more algorithms as the standard.

4. **Should a backup be selected?***
Although only 18 voted for a backup, it should be noted that the use of a backup was not an option specifically listed on the survey. Had it been listed, there may have been more votes. Many people at the conference did not oppose a backup a backup provided that the primary would be required unless it was found unsuitable. Therefore, this issue was not clearly resolved. Such a backup might be used to mitigate the effects of a security attack or an intellectual property claim on the primary algorithm.

5. **What is the best key size to use?***
Certain submitters (including some whose algorithms ran equally fast for all size keys) felt that only the 256-bit key should be used. This would provide for a very high level of security right from the beginning. Rijndael, however, uses additional rounds for the larger key sizes and encrypts 40% slower with a 256-bit key than with a 128-bit one. Many applications today support 128-bit keys. Since NIST requested an algorithm that supports three key sizes, it seems appropriate that the FIPS support all three sizes. The implementer can then decide what size key is best for a particular application. The standard should not require that all three keys sizes be implemented.

6. **Should the AES selection be based solely on technical merits rather than political considerations?**
One verbal suggestion that was made at the AES3 conference was that the NIST technical team should announce its decision before high level management can change the decision for political reasons. There also seemed to be some fear that only an algorithm from a U.S. company could win. While the NIST selection team may have to brief higher management on their selection, it is important that the decision remain one based on merit.

7. **What should be the NIST selection?***
   Rijndael was the clear favorite among the conference and it appears to be efficient and flexible over a wide variety of applications. It would be difficult not to select Rijndael without some very strong reason. Unfortunately, its only significant area for concern appears to be security. The Substitution Permutation technology upon which it is based is less well known and the security margin is the lowest of the finalists. NIST will have to decide whether this is a major concern. If the rounds are not increased, then the algorithm should be specified in a manner so that implementers may easily increase the rounds in the event that it is necessary at a later time.

   Serpent would be a conservative alternative. While it also uses the Substitution Permutation technology, most felt that it offered the highest level of security. However, Serpent did not appear to be as flexible as Rijndael and tended to be slower in software.

   If a runner-up is selected, Serpent or Twofish seem like good candidates. They both have large security margins and are likely to remain secure for quite some time. Both MARS and RC6 had negative scores as described in issue 2 of this section. Clearly, MARS was perceived to be too complex and RC6 was perceived to be too PC oriented. Only time will tell for sure whether these perceptions were correct.

8. **If a backup is selected, how should it be used?***
   If a backup is selected, most of the AES3 participants felt that the backup should not be used unless NIST found a major flaw with the primary. The backup should be described in the AES FIPS where it would be explained that initially only the primary was to be used. This would give NIST the option of quickly moving to the backup if needed at a later time. Since it would take NIST years to make a new standard, this would save considerable time. Even if the backup was not the ultimate replacement, it could be used as an interim much like triple DES is being used as an interim today. What needs to be considered is whether vendors at their option could implement the backup along with the primary. This would permit a very quick change-over for equipments implementing both algorithms. In many cases users would be willing to pay extra for cryptography which can convert to a backup algorithm if necessary. The ANSI Accredited Standards Committee, X9, has already stated that it would like to be able to replace the cryptographic algorithms in its financial systems. Such a backup system would give that capability.

9. **Should NIST specify the bit ordering of input parameters in the standard?**
   There seems to have been confusion over bit ordering with DES and also with some AES specifications. It was mentioned at the conference that NIST should take special measures to ensure that this confusion does not continue when the AES is published.

10. **Should NIST be responsible for maintaining the standard (i.e., encouraging additional analysis)?**
    Some mentioned that the study of the AES should continue even after it is selected and becomes a FIPS standard. The cryptographic community will continue to study AES because it will be a U.S. government standard. NSA also continues to study FIPS cryptographic algorithms and a formal review is conducted every five years.

11. **What modes of operation should be developed?***

    Clearly, the basic four modes originally specified for DES should be permitted for AES. In addition, several of the conference attendees expressed the need for additional modes that can encrypt blocks in parallel thereby permitting very fast encryption. NIST should consider ANSI X9.52, which defines several interleaved and pipelined modes which meet this requirement. Some also expressed a desire for a counter mode. NIST should consider the PCBC modes specified in [4] and the stateful and probabilistic modes defined in [5] to provide data authentication modes of encryption. NIST should also define a 256-bit block mode so that the AES could be used as a hash function.

12. **What process should NIST use to develop the Modes?**
    Those at the AES3 conference felt that NIST should hold another conference to discuss AES modes of operation. This would require that NIST call for papers beforehand and provide for sufficient research time.

13. **How closely should NIST follow the survey results?***
    The research results express the views of the majority of those attending the AES3 conference. Clearly, they have a strong interest in the AES. But the group tends to be weighted in favor of cryptographers and implementers over users. NIST must take into consideration the views and needs of all parties. For example, government users were not widely represented. Nevertheless, since NIST asked for public participation and held the conference, the opinions of the AES3 attendees should carry significant weight. Any choice which varies radically from the views of the attendees would require considerable justification.

References:

1. A Strategy for Analyzing Public Comments and Preparing the Round 2 Status Report, Miles Smid, NIST Purchase Order 43SBNB067018, May 22,2000.

2. RC6 as AES, Ronald L. Rivest, M. J. B. Robshaw, and Yiqun Lisa Yin, Proceedings of the Third AES Candidate Conference, National Institute of Standards and Technology, April 13-14, 2000, (see AES3 home page for an electronic version of this paper).

3. Bitslice Ciphers and Power Analysis Attacks, Joan Daemen, Michael Peeters and Gilles Van Assche, Preproceedings of Fast Software Encryption Workshop 2000, New York, New York, April 10-12.

4. Integrity-Aware PCBC Encryption Schemes, Virgil Gligor and Pompiliu Donescu, 7th International Workshop on Security Protocols, Cambridge, U.K., April 1999, August 6, 1999.

5. Unforgeable Encryption and Chosen Ciphertext Secure Modes of Operation, Johathan Katz and Monti Young, Preproceedings of Fast Software Encryption Workshop 2000, New York, New York, April 10-12.